Correlated Power Noise Generator as a Low Costs DPA Countermeasures to Secure Hardware AES Cipher

Najeh Masmoudi Kamoun, Lilian Bossuet, Adel Ghazel

Correlated Power Noise Generator as a Low Cost DPA Countermeasure to Secure Hardware AES Cipher

Najeh Kamoun[1], Lilian Bossuet[2], and Adel Ghazel[1]
[1]CIRTA'COM, SUP'COM
Tunis, Tunisia
najeh.kamoun@insat.rnu.tn, adel.ghazel@supcom.rnu.tn
[2] IMS, University of Bordeaux
Bordeaux, France
lilian.bossuet@ims-bordeaux.fr

Abstract — **To secure cryptography hardware implementation many works are focusing on side-channels attacks. For such attacks, several different countermeasures can be done at different levels abstraction. But all published countermeasures lead to a significant area and power consumption overhead. In this paper, we present a new countermeasure against DPA attack which also leads to very small implementation compared to existing countermeasures such as the most used: masking schemes. The proposed approach is to use a correlated power noise generator to remove the design power correlation with the secret key. Its efficiency is proved with a practical DPA attack realization on Actel Fusion FLASH FPGA and Xilinx Virtex 4 SRAM FPGA. With the proposed countermeasure, the full 128-bits AES implementation on Xilinx Virtex 4 has a smaller area overhead (12.78 times less) than masking scheme countermeasure.**

Keywords: DPA, AES, FPGA, countermeasures, hardware securiy.

I. INTRODUCTION

The Advanced Encryption Standard (AES) [1] will replace the old DES algorithm [2] in a broad variety of applications. Now, AES has become a world-wide standard with an even wider use than the DES. During the AES selection process, the security of AES was evaluated with respect to all types of attacks.

While being resistant to the classical cryptanalytic methods, it turned out that so-called implementation attacks are a serious threat against naive implementations of the AES algorithm. Implementation attacks refer to a new class of cryptanalysis methods, which are aimed against implementations of cryptographic primitives. Power analysis attacks are passive non invasive attacks. Kocher et al. have shown they are very effective and relatively cheap to conduct in practice [3].

In this paper, we will focus on Differential Power Analysis DPA [3]. In this attack, an adversary predicts target device power consumption with an hypothetical model. Those predictions are then compared to the real, measured power consumption in purpose to recover secret key. DPA has been shown efficient to defeat processor, ASIC and even FPGA implementations [4-5].

To overcome this attack, countermeasures can be implemented to secure AES implementations. They can be at the gate, system, or algorithmic levels. The first ones are based on the logic styles that aim to achieve independence between secret key and the power consumption. The second ones focus on into deteriorating side-channel signal quality. The third ones mask the manipulated data to remove this dependence. At the gate level, they are the most costly for area and power consumption. In embedded design case, all this countermeasures are too area and power consuming. To answer to this issue, we propose a new low-cost architectural countermeasure.

In this paper, we investigate the possibility to counteract these attacks by using a new correlated power noise generator. This new countermeasure name is "*interfering countermeasure*". Its basic idea is to add an interfering power signal which depends on the manipulated data and an interfering key. This key value has to be different from the secret key value.

Paper is organized as follows. In the second section we present some interesting previous works. We explain, in section 3, our new approach to secure AES implementation based on power interfering. Section 4 gives the experimental results that permit us to conclude to the countermeasure efficiency. In section 5, we discuss the FPGA implementation performance comparison between our solution and the masking scheme. Based on these results we draw some conclusions in section 6.

II. PREVIOUS WORKS

Power analysis attacks work because the cryptographic devices power consumption depends on the executed cryptographic algorithms intermediate values. Hence, countermeasure goal against such attacks is to make the power consumption independent of those intermediate values. Several ones are suggested in the literature [4]. At the security point of view, the more efficient countermeasures acts at the algorithmic level or at the logic level.

Algorithmic countermeasures can be restricted to the masking schemes. Its basic idea is to randomize the intermediate results that are produced during the computation of a cryptographic algorithm [6]. Such solutions need some modification in the design conception and a True Random Number Generator (TRNG) to generate the random masked [7]. Masking scheme is a stronger countermeasure against first order DPA. Nevertheless this countermeasure area overhead could be significant as we have shown in [8]. Moreover, such countermeasure is sensible to the glitch issue. It is inefficient if

the attacker obtain some post-layout information as demonstrated in [9].

Logic countermeasures are based on special leakage-resistant logic styles [10-11]. The power consumption of a circuit depends on the output transitions of the gates. If these transitions depend on secret information, the security of the implementation can be compromised. Hence, to make a circuit resistant against power analysis attacks, the power consumption should be independent of the secret information. The circuit-level approaches to achieve this can be divided into two categories: custom logic styles and standard logic styles. Custom logic styles are only applicable to custom ASIC implementations. Standard logic styles combine standard cells from existing libraries into new standard cells. In most of all cases these countermeasures are glitch sensitive. They need particular design of balanced architecture. Moreover such countermeasures are very area and power consuming [10-11].

In [12], Standaert *et al*. introduce architectural countermeasure. Their idea is to generate an additive power noise signal from the part of the AES blocs by using unrolling architecture. As the AES rounds input are uncorrelated, each round is a power noise generator for the other rounds point of view. Add a power noise to the instantaneous power consumption feel an interesting way to protect the cipher against the DPA. Nevertheless such countermeasure is inefficient. Actually, the averaging in DPA filters out uncorrelated noise from the differential power trace. With such countermeasure it is only arduous to perform the DPA but always possible. The main advantage of architecture countermeasure is the very low area cost and low power consumption without frequency decrease.

In order to take benefit of the architectural countermeasure with a high level of security, in the next section, we will introduce a new architectural countermeasure which uses a correlated power noise generator.

III. PRINCIPLE OF INTERFERNCE COUNTERMEASURE

In hardware implementation of AES, during the *SubBytes* step the power consumption traces are correlated with data computed in the design. These computing data are correlated with the AES secret key *K* used in the design. It seems like that *K* is printed in the power consumption traces. To be very quick and very efficient, DPA attacks take advantage of this security failure named *Side Channel*.

To eliminate the side channel, our idea is to interfere the AES cipher power signal with a power signal correlated with cipher input data D_{in} and with an interfering random key K_{interf}. This architecture level countermeasure is illustrated by figure1. The input data D_{in} is inserted simultaneously to two encryption cores: a classical AES core and an interference core used like a *Correlated Power Noise Generator*. The AES core performs the first *AddRoundKey* step with cipher data input D_{in} and with the secret key *K*. At the same time, in the interference core, the cipher input data D_{in} is provided to a similar module *AddRoundKey* but with the interfering key K_{interf}. The outputs of the two *AddRoundKey* modules are applied to two similar *SubBytes* modules synchronously. The *SubBytes* modules outputs signals are two signals *S* and S_{interf}.

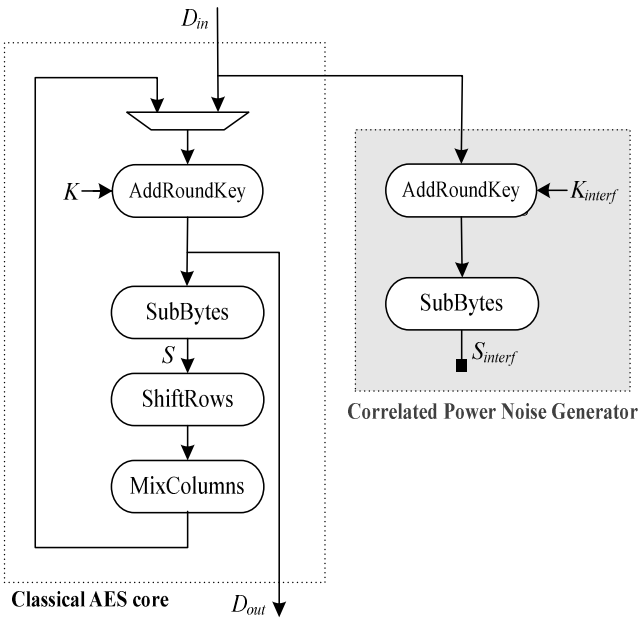


Figure 1. Description of the new interference countermeasure.

Like the signal S switching, the signal S_{interf} switching is correlated to D_{in}. As the global power consumption is due to S and S_{interf} switching both it is not only correlated to the secret *K* and the data D_{in}. Actually, the global cipher power consumption is corralled to data D_{in} and the couple of secret keys (K, K_{interf}). Thereby, it is not possible to extract the secret key *K* by the classical DPA attacks as we will experimentally prove in the next section

Nevertheless, to be a DPA efficient countermeasure, the only condition needed is to have different value for secret key *K* and interference secret key K_{interf} as describe in the following rule:

$$K_{Interf} \neq K, K_{Interf} \in [0..255]$$

To formally prove the benefit of the proposed countermeasure we use the Guilley et al. work [13]. In this work the authors give the first equation (1) that provides the formal justification to the DPA attacks with correlation analysis:

$$E\left(P_{th} \times \left(-2.\sum_{j\in J}(-1)^{j(t+1)}\right)\right) = \frac{\sum_{j\in J}\xi_j^{\uparrow} - \xi_j^{\downarrow}}{2} \qquad (1)$$

Where P_{th} is the chip power consumption, $(-1)^{j(t+1)}$ is the balanced hamming weight, *J* is the set of nets in the netlist, $\xi^{\uparrow}$ and $\xi^{\downarrow}$ the fall and rise transition power of a single net. This equation main assumption is that the manipulated data in the design are independent. In our case, we assume that the manipulated data are dependent. In conclusion, with our

interfering countermeasure the equation (1) is not valid. So it is impossible to recover the secret key K with classical first order DPA without cipher text use.

In the following section, we will experimentally prove the validity of this conclusion.

IV. EXPRIMENTAL VALIDATION AND RESULT

A. DPA experimentation

1) Methodology of validation

For DPA attack, the correlation analysis is chosen to evaluate the secret key. It consists in using Pearson coefficient between the measured power consumption V_M and the power model M_P. In our work, we use the hamming weight for the 8 bit output of the *SubBytes* function.

$$C(V_M, M_P(c_i)) = \frac{E(V_M . M_P(c_i)) - E(V_M).E(M_P(c_i))}{\sqrt{\text{var}(V_M).\text{var}(M_P(c_i))}} \quad (2)$$

To validate our countermeasure, we used two experiments. The first demonstrates that with independent inputs the DPA attack is still achievable. The second shows unsuccessful DPA practical implementation in the secure design with the proposed countermeasure.

Before these two experiments, we have first attacked a sample design witch only contain the two first steps of an AES cipher (*AddRoundKey* and *SubBytes*). The experimental setup of the practical implementation of DPA attack will be described in the following. This first step has permitted us to validate our DPA lab described in the Figure 2.

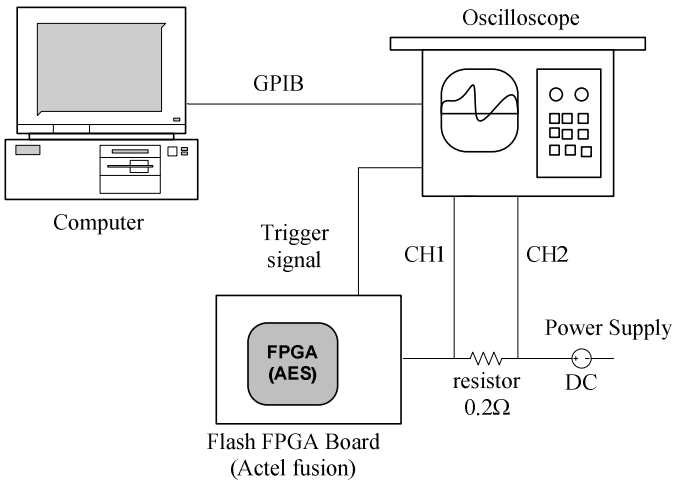


Figure 2. Exprimental lab for DPA attack

We employ an Agilent 54622D digital oscilloscope which has a bandwidth of 100 MHz with maximum sampling rate of 200 MegaSample/sec. To obtain enough sample points per cycle, we lowered our design speed to 95 Hz.

The communication between the scope and the PC is done via the General Purpose Bus Interface (GPIB) which is specified in IEEE-488 standard. We insert a resistance with a 0.2 Ω value between the power supply and the FPGA Board. We measure the voltage difference between CH1 and CH2. Note that the probe used has a low pass frequency response that will reject any DC signal, but will pass variable AC signals with bandwidth below to 20MHz.

The next section will present and discus the obtained experimental results.

2) DPA attack of unsecured AES S-Box

To first test the DPA lab with power correlating analysis, we implemented a sample design with an ACTEL Fusion AfS600-FG256 FLASH FPGA, and with a XILINX Virtex4 XCVLX25 SRAM FPGA.

This sample design in figure 3 includes the 8-bits hardware module *SubBytes* and *AddRoundKey*. The input data D_{in} is first added with a secret sub-key (8 bits) K with exclusive-or before to input the *SubBytes* module. To design the *SubBytes* module, we use the smallest S-Box up to date introduced by Canright [14] and we implement it [15].

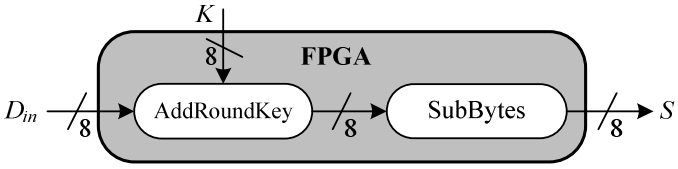
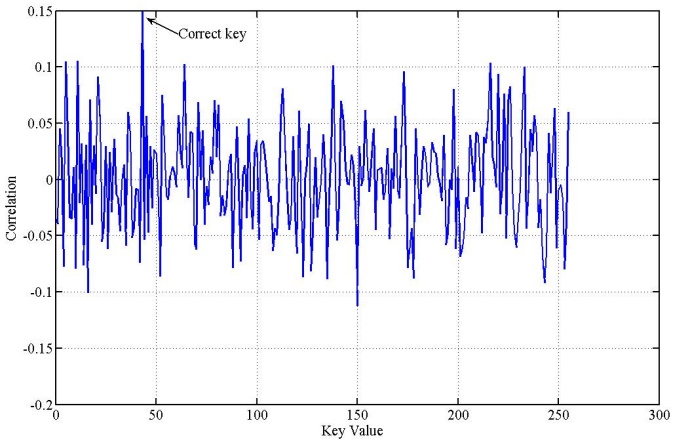


Figure 3. Attacked sample design block diagram.

Figure 4. Successful attack on the first two AES rounds for the correct key



K=43 with 1000 measurements.

With such DPA experimentation, figure 4 shows a successful correlation attack result with an unsecure cipher which uses the secret key K=43 with 1000 measurements. That confirms the DPA lab efficiency.

B. *DPA Attack on two adjacent S-Box with independent input data*

To validate our countermeasure, we first apply simultaneously two independent input data D_{in1} and D_{in2} to two *AddRoundKey* modules. These modules use two 8-bits distinct secret keys K_1 and K_2. Figure 5 shows the attacked design. As show by this figure, one part (driven by D_{in2}) of this dual architecture could be considered as an uncorrelated power noise generator for the other part (driven by D_{in1}).

We employ the same experimental DPA setup as explained in the previous subsection. The keys K_1 and K_2 are extracted with correlation power analysis as we can see on figure 6 (K_1 successful DPA attack) and figure 7 (K_2 successful DPA attack). The results are given with Actel Fusion FLASH FPGA,

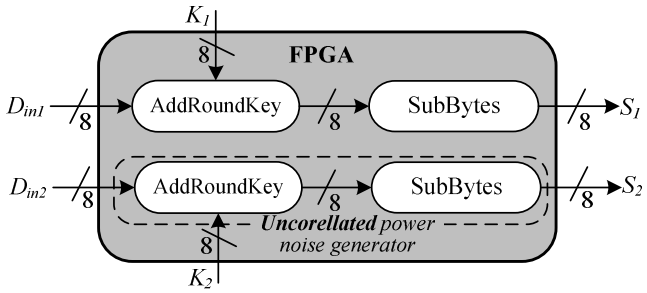


Figure 5. Attacked sample design with **uncorrelated** power noise generator.

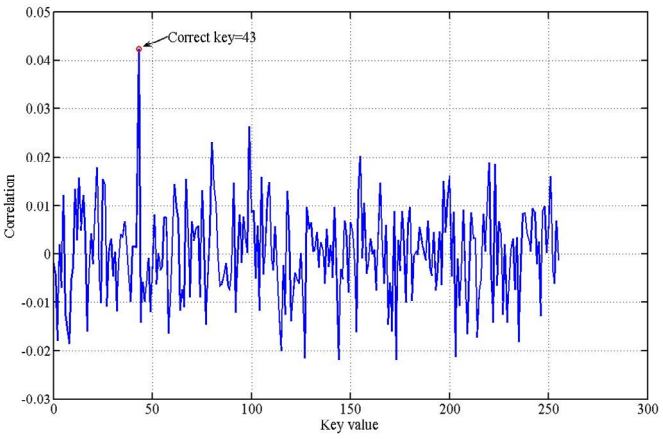


Figure 6. Successful K_1 attack with **uncorrelated** power noise generator (K_1=43 and K_2=145), 12 000 measurements.

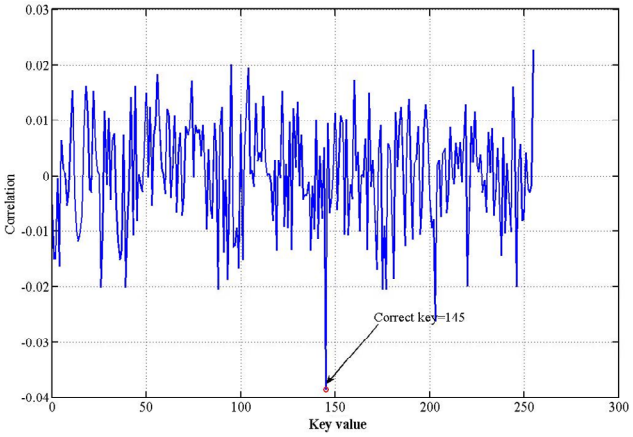


Figure 7. Successful K_2 attack with **uncorrelated** power noise generator (K_1=43 and K_2=145), 12 000 measurements.

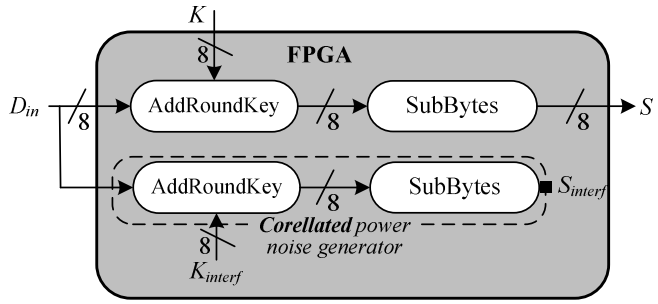


Figure 8. Attacked sample design with **correlated** power noise generator (proposed DPA countermeasure).

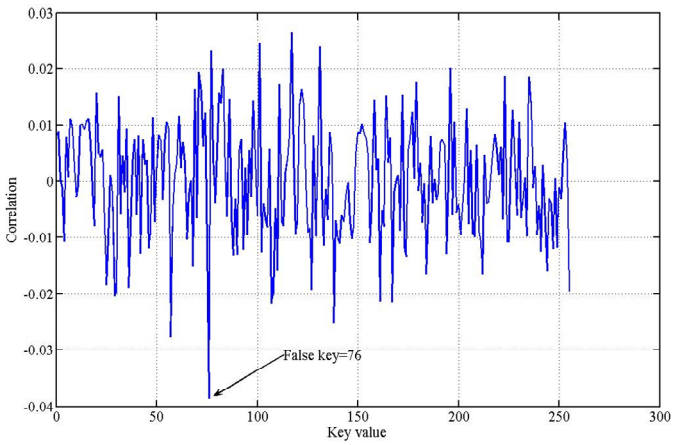


Figure 9. Unsuccessful K attack with **correlated** power noise generator (K=43 and K_{Interf}=145), 12 000 measurements.

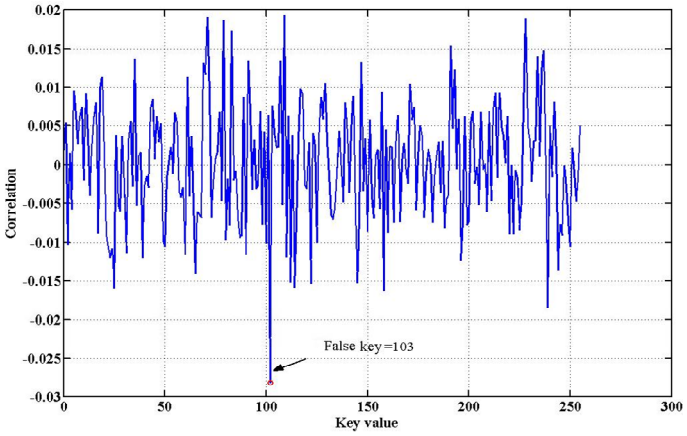


Figure 10. Unsuccessful K attack with **correlated** power noise generator (K=43 and K_{Interf}=145), 20 480 measurements.

however there are similarly testing with Xilinx Virtex4 SRAM FPGA.

As a conclusion we obtain the same results that Standaert's work, the DPA filters out the uncorrelated power noise. An additive uncorrelated power noise is not an efficient DPA countermeasure DPA attack on secure AES S-Box with interference countermeasure.

1) DPA attack without considering the structure of the interference countermeasure

For the last experiment we use the proposed countermeasure. In this case, to generate a correlated power noise, the same input data D_{in} is used for the two modules (*AddRoundKey* and *SubBytes*) as figure 8 shows. So, as show by this figure, one part (which uses the interference secret key K_{interf}) of this dual architecture could be considered as a correlated power noise generator for the other part (which uses the interference secret key K).

Like for the validation test, we use 12 000 power traces measurements. The DPA doesn't permit us to detect the secret key K as is shown in figure 9.

We increase the number of measurements up to 100 000. We do not detect the secret key K. Figure 10 illustrates the result of unsuccessful attack with 20 480 measurements.

We have tested with all possible values (255 values corresponding to 2^8 values space without the secret key value) of interference 8-bits secret key K_{interf}. The results are the same with all the possible keys.

2) DPA attack with considering the structure of the interference countermeasure

The structure of our interference countermeasure is taking in consideration for DPA attack. Indeed, it possible to take into account of the countermeasure by changing the DPA power consumption model *PM*:

$$PM=\mathrm{f}(\mathbf{H}(S)+\mathbf{H}(S_{Interf})) \quad (2)$$

Where **H** is the Hamming weight function, S and S_{Interf} are signals mentioned in figure 1. They are respectively the output of *SubBytes* with D_{in} as input and the useful key K and the interference key K_{interf}.

We suppose that we doesn't know the interference key K_{interf}, we realize the DPA attack on our design with considering the power consumption model *PM* (equation (2)). The result illustrated by figure 11 shows an unsuccessful attack for our design with 20480 traces.

C. Conclusion

We are able to realize a successful DPA attack against an unsecured design with FPGA flash-based and SRAM-based implementation. The same experimental setup is used to try to attack a secure design with the proposed countermeasure. The results are shown that DPA doesn't success to find the secret key K. Such experimentation shows the proposed countermeasure robustness.

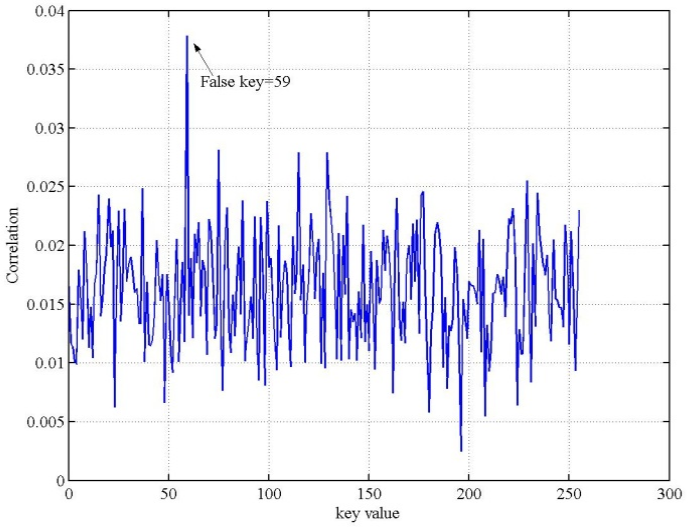


Figure 11. Unsuccessful K attack with new DPA prower prediction model on secure design with the proposed countermeasure (K=43 and K_{Interf}=145), 20 480 measurements.

V. COMPARISON BETWEEN THE PERFORMANCE OF MASKING SCHEME AND THE PROPOSED COUNTERMEASURE

The proposed countermeasure uses the two operations *SubBytes* and *AddRoundKey* with 8-bit data. To compare its performance to the masking scheme, we implement the two operations with three ways: unsecured small S-Box proposed by [14], secure small S-Box with masking scheme proposed by [6] and secure small S-Box with our proposed interference countermeasure. Table I gives the implementation results with a Xilinx Virtex4 SRAM FPGA. Note that a Virtex4 slice is composed by two inputs Look Up Table, two D-Flip-Flop and one carry chain). All results are obtained by using Xilinx 8.2 ISE CAD tool version with default synthesis and place-and-route options.

This table clearly shows that the proposed countermeasure with interference method is very low area cost. The area overhead is only 16 Virtex4 slices for *AddRoundKey* and *SubBytes* implementation. It costs an area around 4 times smaller than the masking solution (without the TRNG area cost).

Unlike the masking solution, with our solution the critical path is not lengthened. Consequently, the maximal frequency for unsecure design and secure design with our proposed countermeasure is the same, whereas the maximal frequency is reduced with masking countermeasure.

We implemented the whole AES on the same FPGA. We use the 16 times S-Box for all the design. Table II summarizes the performance of the different implementations. We note that the implementation of the secure AES with masking do not take into consideration the implementation of the random generator to produce the mask (the same masks are used in all rounds). The area overhead of our secured implementation compared to the unsecured version is less than 5%. The masking solution gives an area overhead of about 60%.

Table I Performance Compraison between AES S-Box implementation in virtex 4 XC4VLX25-FF676

Performance	**AES S-Box**		
	Unsecured [14]	***Secure with Masking* [6]**	***Secure with proposed method***
Area (slices)	36	100	**52**
Area overhead	0%	+ 170%	**+ 44%**
Frequency (MHz)	184	122	**184**
Frequency decreasing	0%	- 33%	**0%**

Table II Performance compraison between implementation full AES in Xilinx Virtex 4 XC4VLX25-FF676 device

Performance	**AES**		
	Unsecured [14]	***Secure with masking* [6]**	***Secure with proposed method***
Area (slices)	1424	2281	**1491**
Area overhead	0%	+ 60,1%	**+ 4.7%**
Frequency (MHz)	143	97	**143**
Frequency decreasing	0%	-11%	**0%**

VI. CONCLUSION

In this paper, a new architectural countermeasure for DPA attacks against AES is introduced. This new low cost solution uses an efficient correlated power noise generator. As the power noise is correlated with the input data it is not possible to filter out it during DPA process. So this solution is DPA resistant as shown by our experimental result. Although, we have not test it with high order DPA, we assume that our solution is the lower cost first order DPA countermeasure. Indeed, some FPGA implementation results, with Xilinx Virtex4 device, have shown that comparing to existing masking scheme and secure logic countermeasure, our solution consume less area and power. Moreover, with an appropriate design, the proposed solution costs any frequency decrease. With the targeted FPGA, the secure AES cipher consumes only 5% more slice than the unsecure cipher.

In the near future, we have to improve the proposed solution to secure the cipher against DPA on the last AES round. Such attack use encrypted texts as a prediction power consumption model input. Actually, the proposed countermeasure architecture (figure 1) could only prevent DPA against AES first round. Nevertheless, if the AES block encrypt output are never output of the chip, that is the case of many application, our solution is sufficient.

REFERENCES

[1] National Institute of Standards and Technology (NIST). FIPS-197: *Advanced Encryption Standard*, November 2001.

[2] National Institute of Standards and Technology. FIPS 46-3: Data Encryption Standard, October reaffirmed 1999.

[3] P. Kocher, J. Jaffe, B. Jun, *Differential Power Analysis*, in the proceedings of Cypto 1999, Lecture Notes in Computer Science, vol 1666, pp 398-412, Santa-Barbara,USA, August 1999, Springer-Verlag.

[4] S. Mangard, E. Oswald, T. Popp, *Power Analysis Attacks: Revealing the Secrets of Smart Cards*, Springer, 2007.

[5] S. B. Örs, E. Oswald and B. Preneel, *Power-Analysis Attacks on an FPGA--First Experimental Results*. Proceedings of Cryptographic Hardware and Embedded Systems – CHES 2003, 5th International Workshop Cologne, Germany, September 8–10, 2003, pp. 35-50, LNCS 2779.

[6] D. Canright and L. Batina, *A Very Compact "Perfectly Masked" S-Box for AES*, Applied Cryptography and Network Security, ACNS 2008, June 3-6, New York.

[7] V. Ficher and M. Drutarovsky, *True Random Number Generator Embedded in a Reconfigurable Device*. In the Proceedings of the 4th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2002), Springer-Verlag, 2002, LNCS 2523, pp. 415-430.

[8] N. Kamoun, L. Bossuet, A. Gazel. *SRAM-FPGA Implementation of Masked S-Box Based DPA Countermeasure for AES*. In the Proceedings of the IEEE International Design and Test Workshop, IDT 2008, Monastir, Tunisa, December 2008.

[9] S. Mangard, N. Pramstaller, E. Oswald, *Successfully Attacking Masked AES Hardware Implementations*. In the Proceedings of the 7th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2005), Springer-Verlag, 2005, LNCS 3659, pp. 157-171.

[10] K. Tiri, M. Akmal, I. Verbauwhede, *A Dynamic and Differential CMOS Logic with Signal Independant Power Consumption to Withstand Differential Power Analysis on Smart Cards*. In Proceedings of the IEEE 28th European Solid-State Circuits Conference (ESSCIRC 02), 2002, pp. 403-406.

[11] T. Popp, S. Mangard, *Masked Dual-Rail Pre-Charge Logic: DPA-Resistance without Routing Constraints.* In the Proceedings of the 7th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2005), Springer-Verlag, 2005, LNCS 3659, pp. 172-186.

[12] F.-X. Standaert, S.B. Ors, B. Preneel, "*Power Analysis of an FPGA Implementation of Rijndael: Is Pipelining a DPA Countermeasure*?", in the proceedings of CHES 2004, Lecture Notes in Computer Science, vol 3156, pp 30-44, Cambridge, MA, USA, August 2004, Springer-Verlag.

[13] S. Guilley, Ph. Hoogvorst, R. Pacalet and J. Schmidt. *Improving Side-Channel Attacks by Exploiting Substitution Boxes Properties*. International Conference on Boolean Functions: Cryptography and Applications (BFCA), may 2007.

[14] D. Canright, *A Very Compact S-Box for AES*, In the Proceedings of the 7th International Workshop on Cryptographic Hardware and Embedded Systems (CHES 2005), Springer-Verlag, 2005, LNCS 3659, pp. 441-445.

[15] N. Kamoun, L. Bossuet, A. Gazel. *Experimental Implementation of DPA Attacks on AES Design with Flash-based FPGA Technology*. In the Proceeding of the Sixth IEEE International Multi-Conference on Systems Signals and Devices, SSD 2009, Djerba, Tunisa, March 2009.